

**Vysoká škola báňská – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra telekomunikační techniky**

**Absolvování individuální odborné praxe**  
**Individual Professional Practice in the Company**

**2014**

**Lukáš Janoška**

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra telekomunikační techniky

## Zadání bakalářské práce

Student:

**Lukáš Janoška**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R059 Mobilní technologie

Téma:

**Absolvování individuální odborné praxe  
Individual Professional Practice in the Company**

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: InfoMED s.r.o.
2. Struktura závěrečné zprávy:
  - a. Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
  - b. Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti
  - c. Zvolený postup řešení zadaných úkolů
  - d. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
  - e. Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
  - f. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Zdeňka Chmelíková, Ph.D.**


Konzultant bakalářské práce: Ing. Roman Míkula

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014

  
doc. Ing. Miroslav Vozňák, Ph.D.  
vedoucí katedry

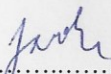


  
prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## Prehlásenie študenta

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave dňa: 2. mája 2014

  
.....  
podpis študenta

## **Pod'akovanie**

Rád by som poďakoval Ing. Zdeňke Chmelíkovej, Ph.D. a Ing. Romanovi Mikulovi za odbornú pomoc a konzultácie pri vytváraní tejto bakalárskej práce. Ďalej by som rád poďakoval Ing. Pavlovi Sečníkovi za umožnenie vykonávania praxe v jeho spoločnosti.

## Prohlášení zástupce spolupracující právnické nebo fyzické osoby

„Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.“

Dne: 2. mája 2014



**infoMED s.r.l.**

IČO: 36 317 381  
Stred 57/51-8 ③

017 01 POVAŽSKÁ BYSTRICA

podpis zástupce

## **Abstrakt**

V bakalárskej práci popisujem priebeh vykonávania odbornej praxe, ktorú som absolvoval vo firme InfoMED s.r.o. V úvode práce uvádzam profil firmy a jej zameranie. Popíšem čo bolo náplňou mojej praxe a ako som bol firme užitočný. Ďalej popíšem podrobné riešenie zadaných úloh, aké technológie som pri tom použil a na záver uvediem znalosti získané behom štúdia, ktoré som pri práci uplatnil.

## **Kľúčové slova**

Odborná praxe; Android; Android SDK; Java; SQL; IntelliJ IDEA; Web platform; JSON; OData; odata4j; Servisný tiket; Objednávka; Jazda

## **Abstract**

In this bachelor thesis I describe my experiences during individual professional practice at the software company InfoMED Ltd. At the beginning of the thesis I have introduced the company's profile and its focus. I have described what my main job was and how useful I have been. I describe in detail solutions of assigned tasks, what technologies I have used and at the end I mention knowledge learned at school which I have find useful during my professional practice.

## **Key words**

Professional practice; Android; Android SDK ; Java; SQL; IntelliJ IDEA; Web platform; JSON; OData; odata4j; service report; order; journey

## Zoznam použitých skratiek

Skratka	Význam
<b>IDE</b>	Integrated Development Environment
<b>UI</b>	User interface
<b>PC</b>	Personal computer
<b>Http</b>	Hypertext Transfer Protocol
<b>AWP</b>	Advantage web platform
<b>PDA</b>	Personal digital assistant
<b>OData</b>	Open data protocol
<b>ADS</b>	Advantage database server
<b>AWP</b>	Advantage web platform
<b>TCP</b>	Transmission Control Protocol
<b>UDP</b>	User Datagram Protocol
<b>XML</b>	Extensible Markup Language
<b>JSON</b>	JavaScript Object Notation
<b>CRUD</b>	Create, read, update, delete
<b>JDBC</b>	Java-based data access
<b>ODBC</b>	Open Database Connectivity
<b>SQL</b>	Structured Query Language
<b>SDK</b>	Android Software Development Kit
<b>AVD</b>	Android Virtual Devices
<b>OASIS</b>	Organization for the Advancement of Structured Information Standards
<b>URL</b>	Uniform Resource Locator

## Obsah

1	Úvod.....	- 10 -
1.1	Profil firmy InfoMED s.r.o .....	- 11 -
2	Zadanie a požiadavky od firmy .....	- 12 -
3	Používané technológie .....	- 14 -
3.1	IntelliJ IDEA .....	- 14 -
3.2	Advantage web platform .....	- 14 -
3.3	OData .....	- 14 -
3.4	odata4j .....	- 14 -
3.5	JSON .....	- 15 -
3.6	Android Software Development Kit.....	- 15 -
3.7	Android emulátor .....	- 15 -
3.8	Google Chrome .....	- 15 -
4	Riešenie zadaných úloh.....	- 16 -
4.1	Zvládnutie problematiky servisného tiketu .....	- 16 -
4.2	Zoznámenie so systémom InfoLITE .....	- 16 -
4.3	Zoznámenie s databázou .....	- 16 -
4.4	Testovanie spojenia s databázou .....	- 17 -
4.5	Štúdie výberu vhodných technológií.....	- 17 -
4.6	Vytvorenie prihlasovacej aktivity .....	- 17 -
4.7	Vytvorenie hlavnej aktivity .....	- 18 -
4.8	Editácia servisného tiketu.....	- 21 -
4.9	Vytvorenie servisného tiketu.....	- 21 -
4.10	Vytvorenie novej objednávky a jazdy .....	- 22 -
4.11	Android manifest.....	- 22 -



4.12	Viacjazyčnosť aplikácie .....	- 23 -
4.13	Testovanie aplikácie na reálnych zariadeniach .....	- 24 -
4.14	Ošetrovanie výnimiek, stabilita aplikácie. ....	- 24 -
5	Využitie znalostí nadobudnutých počas štúdia .....	- 26 -
6	Záver .....	- 27 -
	Použitá literatúra .....	- 28 -

# 1 Úvod

Bakalárska práca popisuje absolvovanie individuálnej odbornej praxe v softvérovej firme z pohľadu študenta. Odbornú prax som vykonával vo firme InfoMED s.r.o[1]. Prax som vykonával počas 55 pracovných dní v zimnom a letnom semestri akademického roku 2013/2014. V práci je popísané odborné zameranie firmy InfoMED s.r.o a moje zaradenie do tímu. Ďalšie časti práce sa zaoberajú zadanými úlohami od firmy, použitými technológiami a podrobným popisom riešenia zadaných úloh. V závere popíšem znalosti nadobudnuté počas štúdia ktoré boli kľúčové a bez ktorých by som sa nebol zaobišiel.

## 1.1 **Profil firmy InfoMED s.r.o**

Firma InfoMED s.r.o. bola založená v roku 2001 ako logické pokračovanie spoločnosti, ktorá sa už viac ako 20 rokov zaoberá vývojom softwaru pre potreby klinických laboratórií. Jej hlavný produkt je vývoj a údržba laboratórneho informačného systému LABIS. Spoločnosť InfoMED s.r.o. sa tiež zaoberá servisom medicínskych zariadení spoločnosti Siemens. Pre tento účel si vytvorila vlastný informačný systém na riadenie servisnej činnosti s názvom InfoLITE.

Systém InfoLITE koordinuje HOTLINE zákaznícku linku s prácou technických pracovníkov (servisných technikov). Eviduje servisnú a aplikačnú databázu. Zabezpečuje hlásenia o pravidelných prehliadkach. Vybavuje objednávky náhradných dielov alebo reagentii a spracováva rôzne reporty. Je to v podstate kompletný CRM systém, ktorý spravuje vzťah medzi zákazníkmi a servisnou organizáciou.

## **2 Zadanie a požiadavky od firmy**

### **Zoznámenie so systémom InfoLITE**

Mojou prvou úlohou bolo sa dokonale zoznámiť so systémom a s jeho funkčnými vlastnosťami. Systém mi predstavil môj konzultant, vytvoril mi konto a vyplnil niekoľko vzorových formulárov aby som mal možnosť pozorovať životný cyklus skutočnej zákazky.

### **Zoznámenie s databázou**

Asi najdôležitejším krokom bolo zoznámenie sa s databázou, s tým aké technológie využíva a aké dáta uchováva. Bolo treba pochopiť aké sú väzby medzi tabuľkami a predstaviť si ako databáza funguje ako celok.

### **Testovanie spojenia s databázou**

Musel som overiť či databáza vracia požadované dáta, nastaviť formát vrátených dát a zistiť ako nad tabuľkami robiť dotazy.

### **Štúdie výberu vhodných technológií**

Táto úloha zahrňovala najmä študovanie oficiálnych stránok a dokumentácií, ale aj programátorských fór. Bolo treba zistiť ktoré technológie sú vhodné na použitie a aké sú ich výhody a nevýhody. Ktoré technológie som nakoniec použil popisujem v jednej z ďalších častí [3].

### **Vytvorenie prihlasovacej aktivity**

Ďalšou úlohou bolo vytvorenie úvodnej aktivity, slúžiacej na overenie užívateľa s dôrazom na bezpečnosť prenosu citlivých informácií.

### **Zvládnutie problematiky servisného tiketu**

Oboznámenie sa s tým čo je to servisný tiket a ako s ním ďalej pracovať.

### **Vytvorenie hlavnej aktivity s dôrazom na dynamické používanie fragmentov**

Po vytvorení prihlasovacej aktivity nasledoval vývoj aktivity ktorá bola jadrom celej aplikácie.

### **Editácia servisného tiketu**

Zobrazenie detailných informácií o servisnom tikete a následná úprava niektorých z nich. Aj to bolo jednou z úloh ktorá mi bola zadaná.

### **Vytvorenie servisného tiketu**

Ďalšou požiadavkou od firmy bolo umožnenie užívateľovi vytvorenie nového servisného tiketu. Požiadavkou bolo, aby UI zodpovedalo editácii tiketu kvôli prehľadnosti používania.

### **Vytvorenie novej objednávky a jazdy**

Užívateľ mal mať možnosť vytvoriť aj novú objednávku a jazdu.

### **Viacjazyčné UI aplikácie**

Keďže vo firme pracujú aj anglicky hovoriaci zamestnanci, nevyhnutnou súčasťou aplikácie musí byť možnosť spustenia v angličtine ako aj v slovenčine.

### **Testovanie aplikácie na reálnych zariadeniach**

Systém Android je v dnešnej dobe najrozšírenejším mobilným systémom bežiacim na stovkách zariadení s rôznym hardwarom, preto nie je jednoduché vyvinúť aplikáciu bežiacu bezchybne na akomkoľvek zariadení.

### **Ošetrovanie výnimiek, stabilita aplikácie.**

Asi nikdy sa programátorovi nepodarí napísať na prvý pokus kód ktorý funguje nekompromisne bez chýb za akejkoľvek situácie. Vždy je treba po sebe kód prekontrolovať a ošetriť možné scenáre a to aj také kedy užívateľ nepracuje s aplikáciou tak ako by sám programátor očakával.

## 3 Použité technológie

### 3.1 IntelliJ IDEA

Aplikáciu som vyvíjal vo vývojovom prostredí IntelliJ IDEA[5]. IntelliJ IDEA je Java IDE od spoločnosti JetBrains dostupná v edíciách community ktorá je zdarma a v edícii commercial. Od verzie 12.1 IntelliJ IDEA podporuje Java 8 a UI dizajnér pre Android vývoj. V roku 2010 dostala najvyššie hodnotenie v teste od Infoworld zo štyroch najznámejších programovacích nástrojov v Jave : Eclipse, IntelliJ IDEA, NetBeans a Oracle JDeveloper. Je to IDE ideálne na vývoj aplikácií pre Android.

### 3.2 Advantage web platform

AWP[3] je webová služba od spoločnosti Sybase, ktorá umožňuje „client-less“ prístup k dátam z Advantage database servera[2], z akéhokoľvek zariadenia, platformy, alebo vývojového prostredia, ktoré dokážu spraviť http request. Prístup nie je obmedzený len zo zariadení ako pc/laptop, mobilný telefón, PDA a tablet. Umožňuje prístup z akéhokoľvek súčasného operačného systému zahrnujúceho Mac OS a Android pretože pre prístup nie je vyžadovaný konkrétny klient. Je postavená na OData webovom protokole pre prístup k dátam. Web platform je webová služba zabalená a postavená na samostatnom module Light Apache Web Server. Je napísaná v C a používa Advantage Client Engine na komunikáciu z webového servera k Advantage Database Server. Webová služba môže bežať na rovnakom hardware ako ADS, alebo na inom hardware v inej sieti. Komunikácia medzi webovou službou a ADS je cez Inter-Process Communication, TCP, alebo UDP. Klienti komunikujú s AWP cez http a XML alebo JSON založenom na OData štandarde.

### 3.3 OData

Open Data Protocol[6] je protokol prístupu k dátam pôvodne definovaný Microsoftom. Verzie 1.0, 2.0, 3.0 boli uvoľnené pod Microsoft Open Specification Promise. Verzia 4.0 už bola štandardizovaná v OASIS a bola sprístupnená v marci 2014. Protokol bol navrhnutý tak aby poskytoval štandardný CRUD prístup k zdroju dát cez webové stránky. Je podobný JDBC a ODBC avšak OData nie je limitovaný na SQL databázy.

### 3.4 odata4j

Odata4j[7] je Java framework ktorý implementuje OData protokol pre obe strany klienta (consumer) aj servera (producer).

### 3.5 JSON

JSON[8] je dátový formát nezávislý na počítačovej platforme, určený pre prenos dát, ktoré môžu byť organizované v poliach alebo agregované v objektoch. Vstupom je ľubovoľná dátová štruktúra (číslo, reťazec, boolean, objekt alebo z nich zložené pole), výstupom je vždy reťazec. Zložitosť hierarchie vstupnej premennej nie je teoreticky nijak obmedzená.

### 3.6 Android Software Development Kit

Nástroje pre vývoj aplikácií pre platformu Android sú obsiahnuté v SDK[10], ktoré je dostupné pre všetky hlavné platformy operačných systémov GNU/Linux, Windows i Mac OS. Sada SDK je rozdelená na tri druhy: základná (nástroje nevyhnutné pre vývoj aplikácií), doporučená a plná konfigurácia vývojového prostredia. Obsahuje nástroj pre ladenie programu (debugger), knižnice, emulátor Android zariadenia, dokumentáciu, ukážky kódu a tutoriály.

V práci budem často používať pojmy ako Aktivita, Fragment, TextView, EditText, Spinner, Checkbox, manifest, navigation drawer, action bar atď. Všetko sú to základné komponenty používané pri vývoji android aplikácie a detailnejšie informácie sa môžete dozvedieť z [4].

### 3.7 Android emulátor

Emulátor[9] operačného systému Android je obsiahnutý v Android SDK. Umožňuje kompilovať vytvorené aplikácie bez fyzického zariadenia na desktopovom PC. Pomocou Android SDK a AVD Manažéra je možné konfigurovať voľbu sieťového pripojenia, SD karty atď. a spúšťať jednotlivé virtuálne zariadenia. Väčšina aplikácií sa chová v emulátore rovnako ako na fyzickom zariadení.

### 3.8 Google Chrome

Internetový prehliadač ideálny na testovanie spojenia s databázou pretože „rozumie“ vráteným dátam z ADS a zobrazuje ich v čitateľnej podobe XML.

## **4 Riešenie zadaných úloh**

### **4.1 Zvládnutie problematiky servisného tiketu**

Servisný tiket slúži na zaznamenávanie dát o činnosti servisného technika na určitom laboratórnom stroji. Servisný tiket uchováva detailné informácie o stroji, jeho sériové číslo, kto je majiteľom atď. Ďalej informácie o tom kto servisný tiket vytvoril a z akého dôvodu (pravidelná prehliadka, porucha stroja, atď.). A v neposlednej rade uchováva informácie o tom ako sa pristúpilo k riešeniu problému, či boli použité nejaké náhradné diely atď.

### **4.2 Zoznámenie so systémom InfoLITE**

Tento systém používajú servisní technici a slúži primárne na vytváranie a editáciu servisných tiketov, vytváranie objednávok tovaru a reagencií, zaznamenávanie jázd a dovolení zamestnancov a obsahuje mnoho ďalších pokročilých funkcií.

Po dokončení práce na laboratórnom stroji sú technici povinní v systéme vyplniť servisný tiket. Doteraz mal technik ma dve možnosti. Buď sa pripojiť vzdialene cez smartphone na windows server, avšak rozhranie systému nie je prispôsobené na malý display mobilného zariadenia, alebo so sebou nosiť notebook. Ani jedna z možností nie je veľmi pohodlná a práve tieto problémy mala eliminovať nová mobilná aplikácia, ktorá sa síce funkcionalitou nevyrovná systému InfoLITE avšak beží na mobilnom zariadení s natívnym UI a sústreďí sa na veci, ktoré sa v teréne najčastejšie používajú, čím veľmi uľahčuje prácu servisným technikom.

### **4.3 Zoznámenie s databázou**

Keďže InfoLite pracuje s databázou od spoločnosti Sybase, Advantage database server, bolo treba naštudovať funkčné požiadavky tejto databázy a možnosti vývoja mobilnej aplikácie. Vedúci vývoja mi vytvoril novú testovaciu databázu s plnohodnotnými dátami a ja som sa k nej mohol pripojiť pomocou Advantage web platform. Databáza počúva na sokete, ktorý je definovaný IP adresou, portom a protokolom. Vedúci vývoja mi priradil konto s administrátorskými právami. Mobilná aplikácia komunikuje s rovnakou databázou ako systém InfoLITE a tým je zabezpečená integrita a predchádza sa redundancii dát.



#### 4.4 Testovanie spojenia s databázou

Prvotné testovanie som robil v okne prehliadača. Databáza fungovala dobre, vracala požadované dáta v prípade keď som selektoval všetky atribúty, avšak nastal problém keď som chcel selektovať konkrétne atribúty. Po dlhom a neúspešnom hľadaní riešenia problému som kontaktoval priamo spoločnosť Sybase s otázkou prečo nefunguje funkcia select (extended query options) s konkrétnymi parametrami. Odpoveď prišla okamžite s tým, že bohužiaľ táto funkcionality nie je v súčasnej, prvej verzii dostupná. Rozšírené funkcie budú dostupné v ďalšej verzii databáz, ktorá je plánovaná na Q3 2014. Bolo mi navrhnuté vhodné riešenie mojej situácie v podobe použitia Views s query options ktoré sú v najnovšej verzii dostupné. To znamenalo, že som si dopredu v databáze vytvoril View ktoré reprezentuje konkrétny select nad tabuľkou. Po otestovaní to naozaj fungovalo, takže som si pripravil konkrétne Views a nad nimi robil dotazy.

#### 4.5 Štúdie výberu vhodných technológií

Bolo treba rozhodnúť, akú technológiu použiť na základné operácie nad databázou z Androidej aplikácie. Priamo na oficiálnych stránkach ADS som našiel ukázkový projekt, ktorý využíval odata4j. Tento Java framework som vybral, pretože bol doporučený vývojármi ADS. Použitie nebolo zložité, stačilo stiahnuť distribučný archív a v konfigurácii projektu pridať odata4j-bundle-x.x.jar do build path. Potom som už len v kóde vytvoril novú inštanciu OdataConsumera, využil som metódy tejto triedy na nastavenie URL, prihlasovacích údajov a formátu. Dáta som získal pomocou cyklu foreach kde som prechádzal všetky objekty typu:

```
OEntity dbitem : consumer.getEntities("NÁZOV  
TABULKY").custom("$format", "json").execute()
```

Jednotlivé properties sa dali získať pomocou:

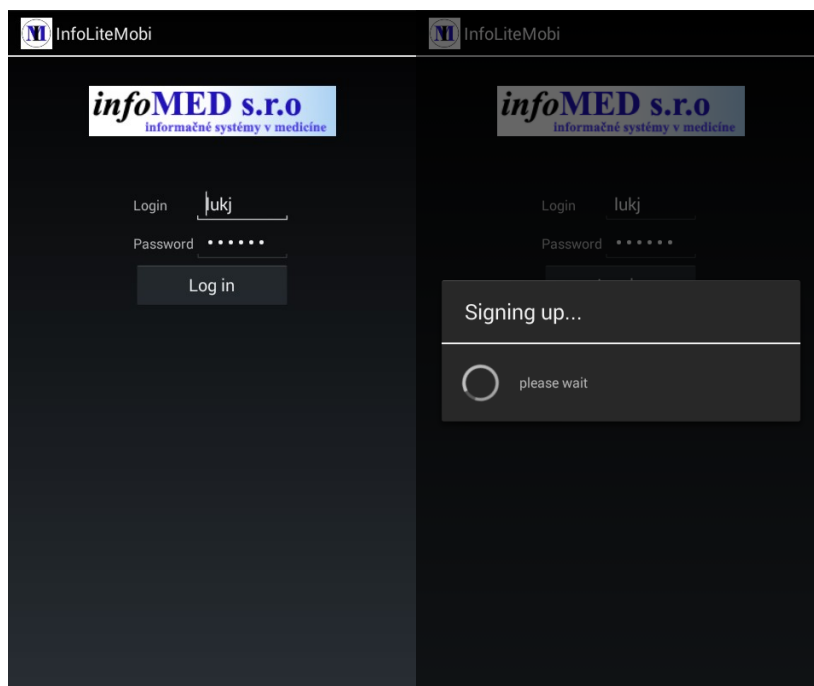
```
user.Email = dbitem.getProperty("email",  
String.class).getValue();
```

Pričom user je inštanciou triedy User. Všetky inšcie aj s ich properties som na záver uložil do ArrayList<TypObjektu> napr.(User, Tiket, Objednavka, Jazda) a mohol som nimi ďalej pracovať.

#### 4.6 Vytvorenie prihlasovacej aktivity

Firma kládla dôraz na to aby sa po spustení aplikácie užívateľovi zobrazila prihlasovacia aktivita, viď obr. 4.1 v ktorej bude zobrazené logo firmy, dve políčka na zadanie

prihlasovacieho mena a hesla, pričom heslo sa nesmie prenášať internetovým spojením bez šifrovania. Riešením tejto úlohy bolo použitie šifrovacích metód a namiesto hesla sa prenáša a porovnáva prepočítaný HASH kód. Po stlačení prihlasovacieho tlačítka sa užívateľovi zobrazí progress dialog oznamujúci overovanie jeho prihlasovacích údajov. Obvykle to netrvá dlhšie ako 2-3 sekundy.



Obrázok 4.1: Prihlasovacia aktivita

### 4.7 Vytvorenie hlavnej aktivity

Po úspešnom prihlásení užívateľa sa spustí hlavná aktivita vid' obr. 4.2 ktorá je najzákladnejším stavebným prvkom aplikácie. V mojej aplikácii som používal Fragments, presne tak ako to najmodernejšie programovanie pre Android vyžaduje a tieto fragmenty boli takmer vždy nejakým spôsobom spojené s hlavnou aktivitou.

Hospital Name	Location	Date	Status	Checkbox
HPL spol. s r.o.	Komárno		archived	<input type="checkbox"/>
BEP III		15:20 14.01.14	closed	<input type="checkbox"/>
HEMO MEDIKA Bratislava, s.r.o.	Bratislava		archived	<input type="checkbox"/>
BEP 2000		11:30 14.01.14	closed	<input checked="" type="checkbox"/>
Národný onkologický ústav	Bratislava		archived	<input type="checkbox"/>
VISTA 1500		10:50 14.01.14	closed	<input checked="" type="checkbox"/>
NTS	Bratislava		archived	<input type="checkbox"/>
Quadriga BeFree		10:30 10.01.14	closed	<input type="checkbox"/>
MEDIREX, a.s.	Bratislava		archived	<input checked="" type="checkbox"/>
BN II		15:00 09.01.14	closed	<input checked="" type="checkbox"/>
NTS	Bratislava		archived	<input type="checkbox"/>

Logged user: *[illegible]*

Obrázok 4.2: Výpis servisných tiketov (landscape mode)

Servisné tikety sú zobrazené v list fragmente, ktorý sa skladá z custom adaptérov umiestnených pod sebou. Custom adapter obsahuje text view s menom nemocnice, mestom, tipom stroja, dátumom a dva check boxy, ktoré zobrazujú či je servisný tiket označený ako archivovaný alebo uzatvorený oba defaultne nastavené na false. V tom prípade je tiket označený ako otvorený. Nové tikety sú archivované = false. Archivované sú tie, ktoré majú archivované = true a museli byť už pred archivovaním označené ako uzatvorené = true.. Užívateľ má možnosť filtrovať tikety pomocou NavigationMode spinneru, ktorý je umiestnený na Action bar. V závislosti na informáciách z check boxov sa filtruje, ktoré tikety sa zobrazia pri kliknutí na ponuku z navigationMode (všetky, nové, archivované).

Po spustení hlavnej aktivity sa v novom vlákne na pozadí začnú automaticky sťahovať nové servisné tikety, pretože tento úkon je najčastejšie používaný servisnými technikmi. Týmto spôsobom som sa pokúsil o vyššiu efektivitu a rýchlosť používania aplikácie. Tikety sa vyfiltrujú pomocou SQL dotazu z pripraveného View v databáze, a tieto sú zoradené podľa dátumu s najnovším servisným tiketom navrchu a zobrazia sa vo Fragmente, ktorý nahradí pred pripravený kontajner pomocou

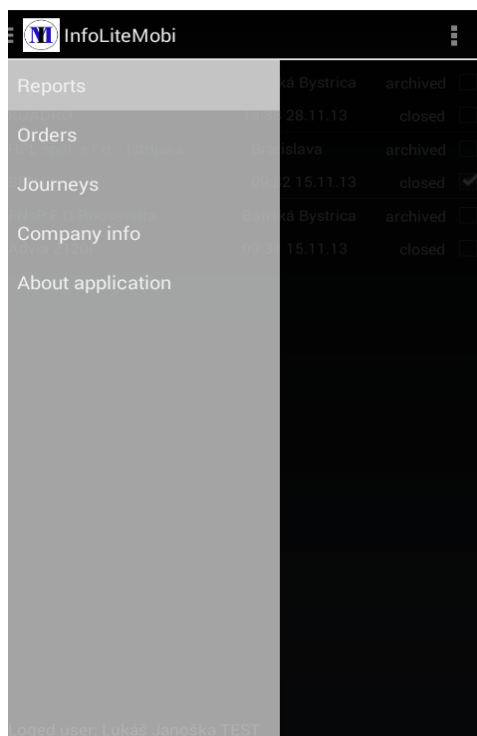
```
android.app.FragmentTransaction ft =
getFragmentManager().beginTransaction();

ft.replace(R.id.fragment_container, lf);
```

```
ft.addToBackStack(null);  
ft.commit();
```

Sťahovanie dát musí prebiehať vo vlastnom vlákne, na to sú citlivé zariadenia s API vyšším 12 a takisto musí prebiehať vo vlastnom fragmente, aby nedošlo k prerušeniu sťahovania dát pri znovu vytváraní aktivity napríklad z dôvodu, že užívateľ otočí display, alebo niekto zavolá atď. Tento fragment je k aktivite pripnutý pomocou funkcie `onAttach()` a to zaručí, že pri znovu vytvorení aktivity z akéhokoľvek dôvodu, fragment dokončí sťahovanie dát a vráti požadované výsledky. Po dokončení sťahovania je zavolaná funkcia `onDetach()` a fragment sa od aktivity odpojí.

V hlavnej aktivite stlačením ikonky aplikácie alebo potiahnutím prsta z ľavej strany obrazovky sa vyroluje Navigation drawer vid' obr. 4.3 fragment, ktorý slúži pre hlavnú navigáciu v aplikácii a umožňuje prepínanie medzi správou tiketov, objednávok a ciest. Ďalej ponúka zobrazenie informácií o firme a aplikácii.



Obrázok 4.3: Navigation drawer

V závislosti na type zariadenia buď hardwarovým tlačítkom menu, alebo tromi bodkami v pravom hornom rohu užívateľ vyvolá ponuku v ktorej je možnosť odhlásenia sa z aplikácie kde sa jednoducho volá metóda `finish()` ktorá ukončí hlavnú aktivitu a tým sa spustí aktivita prihlasovacia.

### 4.8 Editácia servisného tiketu

Každý z `ListFragment`ov zobrazujúcich servisné tikety má implementovanú metódu `onListItemClick()` vďaka ktorej sa po kliknutí na jeden konkrétny tiket zobrazí nová aktivita zobrazujúca všetky informácie o tikete a umožňujúca editáciu tohto tiketu. Použitie fragmentu by v tomto prípade nebolo vhodné z toho dôvodu, že chceme aby sa užívateľ mohol plne sústrediť na vytváranie tiketu aby nedošlo pri tejto činnosti k nechcenému kliknutiu na položky ktoré nie sú nevyhnutne potrebné. Keďže na editáciu tiketu v desktopovom programe InfoLITE bol dostupný monitor klasického PC alebo notebooku nebol tu problém vtesnať všetky potrebné atribúty. S mobilným zariadením je to ale niečo celkom iné a preto som v tejto aktivite použil `ScrollView` teda, rolovateľný kontajner. Nie je žiaduce aby mohol užívateľ editovať základné údaje ako majiteľa, mesto, sériové číslo stroja apod. Má ale možnosť editovať políčka, ktoré hotline centrum nechalo nevyplnené ako napr. spôsob riešenia(výjazd, tel. podpora, podpora), typ výjazdu(pravidelná prehliadka, inštalácia, softwarová aktualizácia atď.), použitý materiál. Všetky tieto možnosti užívateľ vyberá z dopredu pripravených spinnerov. Ďalej má možnosť editovať časy kedy na miesto dorazil a kedy odišiel, tieto pomocou `DatePickeru` kde je defaultne nastavený aktuálny čas. Na záver edituje spôsob eliminácie chyby a má možnosť označiť servisný tiket za uzatvorený. Po stlačení tlačítka aplikácia posiela update operáciu na databázu, ktorá takisto prebieha vo zvláštnom vlákne pomocou `AsyncTask` a vo zvláštnom fragmente.

### 4.9 Vytvorenie servisného tiketu

Na vrchnej lište je umiestnené tlačítko pre vytvorenie nového servisného tiketu. V `layoute` som mu nastavil príznak `android:showAsAction="ifRoom"` takže je vždy zobrazené na hornej lište a nie je schované na rozdiel od tlačítka `logout` ktorému som priradil príznak `android:showAsAction="never"`.

Po stlačení tlačítka na pridanie nového tiketu sa zobrazí nová jednoduchá aktivita v ktorej má užívateľ možnosť doplňovať parametre tiketu do `edittextov`, vyberať zo spinnerov a `datepickerov` a zaškrtnávať `checkbox` uzatvorenia tiketu. Táto aktivita je takisto scrollovateľná a úplne na spodu sú umiestnené dve tlačítka `zruš` a `ulož`(v slovenskej verzii). Po stlačení `ulož` sa

takisto ako pri sťahovaní dát vytvorí nový fragment a v ňom prebieha separátne vlákno a konkrétna update operácia nad databázou s vloženými parametrami prebranými z aktivity vytvorenia servisného tiketu.

Príklad vytvorenia noveho zaznamu:

```
OEntity newProduct = c.createEntity("Products")
    .properties(OProperties.int32("ID", 10))
    .properties(OProperties.string("Name", "Josta"))
    .properties(OProperties.string("Description", "Error
elimination"))
    .properties(OProperties.datetime("ReleaseDate", new
LocalDateTime()))
    .execute();
```

### 4.10 Vytvorenie novej objednávky a jazdy

Celý proces vytvorenia novej objednávky alebo jazdy prebieha na rovnakom princípe ako vytvorenie nového servisného tiketu. Zobrazí sa formulár, kde užívateľ vyplní povinné, prípadne nepovinné položky a po uložení sa vytvorí nový záznam v databáze. Objednávku vytvára technik, ak potrebuje náhradný diel, ktorý práve nemá k dispozícii pri sebe, alebo vo svojom sklade. Naopak jazdu, musí povinne vyplniť technik vždy, keď sadá do vozidla a vyráža na cestu za laboratórnym strojom. Pravým dôvodom vytvárania jazdy je, že je tým technik chránený v prípade nehody. Má dôkaz o tom, že bol vyslaný na cestu, a že nejazdil pre súkromné účely. Ďalším dôvodom je, že hotline centrum vie kde a v akom časovom úseku bude technik a na základe toho mu môže priradiť prípadný ďalší servis v jeho blízkosti.

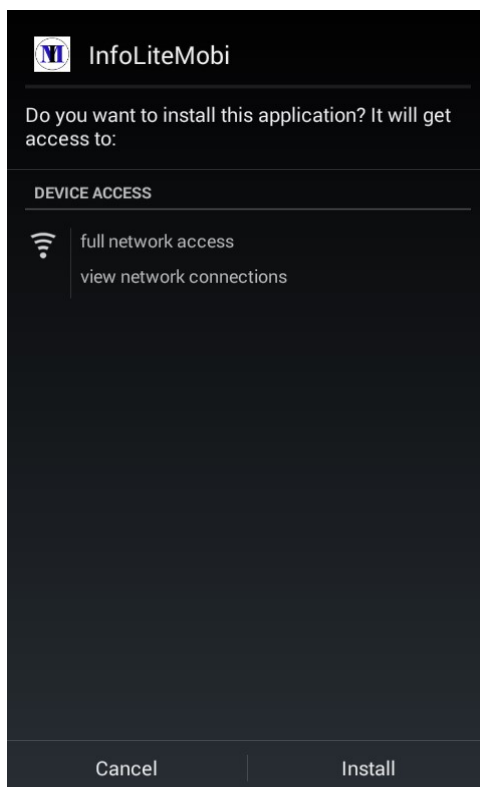
V kóde som sa snažil predchádzať duplicitám, takže niektoré metódy, ktoré som používal pri vytváraní servisného tiketu som zovšeobecnil a tým pádom som ich mohol použiť neskôr pri vytváraní objednávok a jazd.

### 4.11 Android manifest

Aby bolo možné používať internetové spojenie je toto treba v Android manifeste povoliť.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Pri inštalácii aplikácie sa táto pýta vid' obr. 4.4 užívateľa či súhlasí s používaním internetového spojenia.



*Obrázok 4.4: Inštalácia aplikácie*

### 4.12 Viacjazyčnosť aplikácie

Vo firme požadovali aby na prihlasovacej aktivite bola možnosť výberu jazyka. Avšak po diskusií som zadávateľa presvedčil, že najlepším riešením bude využiť štandardný spôsob, ktorý sa používa pri vývoji Android aplikácií a to je spustenie jazykovej mutácie aplikácie v závislosti na lokálnych nastaveniach zariadenia. Je veľmi žiaduce udržiavať všetky reťazce v externých súboroch. Android to robí jednoducho a prehľadne tak, že pri vytváraní projektu pomocou nástroja Android SDK, vytvorí adresár `res/` v najvyššej úrovni projektu. V tomto adresári sa môžu nachádzať ďalšie podadresáre pre rôzne typy zdrojov. Defaultne sa tu nachádza aj `res/values/strings.xml` v ktorom môžu byť uložené reťazce z našej aplikácie. Pre podporu viacerých jazykov bolo treba vytvoriť nový adresár ktorý obsahuje ISO kód krajiny na konci názvu za pomlčkou `res/values-sk` a v ňom nový súbor `strings.xml`, ktorý obsahuje reťazce

preložené do slovenčiny. Android načíta zodpovedajúci súbor na základe lokálnych nastavení zariadenia na ktorom je aplikácia spustená. Ak je systém nastavený na slovenčinu použijú sa zdroje z `res/values-sk/strings.xml`. Ak je systém nastavený na angličtinu, alebo akýkoľvek iný jazyk použijú sa zdroje anglické.

### 4.13 Testovanie aplikácie na reálnych zariadeniach

Pri prvotnom kompilovaní som používal emulátor, ktorý je súčasťou Android SDK, avšak nebol som spokojný pretože emulátor nereagoval vôbec svižne a keďže som nemal k dispozícii procesor na ktorom by sa dala zapnúť virtualizácia bolo kompilovanie na reálnom zariadení jasnou voľbou. K dispozícii som mal Samsung Galaxy S1 pôvodne s Android verziou 2.3.6 neskôr po použití custom ROM vo verziách 4.2.0 a 4.4.2. Na všetkých ROM fungovala aplikácia tak ako sa očakávalo. Testoval som aplikáciu aj na zariadení Samsung Galaxy S3, ktorý má o niečo väčší display, väčšie dpi a rozlíšenie. Takisto aplikácia fungovala dobre a bez akýchkoľvek chýb. To ma ubezpečilo, že kompatibilita je zaručená. Dôležitým krokom bola aj podpora rôznych druhov displayov a použitie vhodných obrázkov a ich rozdelenie do zdrojov podľa dpi na `ldpi` (low), `mdpi` (medium), `hdpi` (high), and `xhdpi` (extra high). Aplikácia slúži len na vnútro firemné použitie preto sa nebude publikovať na playstore ale bude inštalovaná len zamestnancom, ktorí o ňu požiadajú.

### 4.14 Ošetrenie výnimiek, stabilita aplikácie.

Veľmi dôležitou úlohou bolo zaručiť stabilitu aplikácie vo všetkých situáciách. Nesmie sa stať, že aplikácia spadne na nejakej výnimke. Po dôkladnom otestovaní som zistil, že v niektorých netradičných situáciách sa vyskytujú `NullPointerException` napríklad pri prepnutí položky v navigation drawer a následnom okamžitom otočení displayu. Túto výnimku nebol problém odstrániť, stačilo pozmeniť logiku čo bolo pár minút programovania. Ukázalo sa, že dôkladné testovanie je veľmi dôležité a skutočne sa oplatí venovať mu veľkú pozornosť. Avšak hlavné bolo ošetrenie výnimky internetového spojenia. Bolo treba overiť internetové spojenie jednak pri spustení aplikácie a takisto aj pri strate spojenia po spustení aplikácie. Ak došlo k strate spojenia a následnému kliknutiu na položku, ktorá si vyžadovala komunikáciu s databázou užívateľ je upozornený Toastom, štandardným Androidovým nástrojom, ktorý zobrazí v slovenskej verzii hlášku: "Došlo k strate internetového spojenia, pripojte sa prosím na internet." Vid' nasledujúci kód.



```
ConnectivityManager cm = (ConnectivityManager)
getSystemService(CONNECTIVITY_SERVICE);

        NetworkInfo networkInfo =
cm.getActiveNetworkInfo();

        if(networkInfo == null ||
!networkInfo.isConnectedOrConnecting()){
//náš kód napríklad zobrazenie Toastu }
```

## 5 Využitie znalostí nadobudnutých počas štúdia

Pre vývoj aplikácii pre Android sa používa programovací jazyk Java do ktorého základov som prenikol vďaka predmetu PJ I(Programovacie jazyky I).

Veľmi nápomocným mi bol predmet TAMZ II(Tvorba aplikácii pre mobilné zariadenia II), kde som získal prvé skúsenosti s programovaním pre Android. Vďaka tomuto predmetu som sa na praxi nemusel venovať samoštúdiu ale okamžite po obdržaní zadania som vedel ako budem postupovať.

Ďalej by som sa nezaobišiel bez predmetu UDBS (Úvod do databázových systémov), v ktorom som sa naučil ako robiť základné operácie nad databázou, ktoré som nutne potreboval pri získavaní dát z databázy ako aj pri pridávaní nových záznamov alebo editácií existujúcich.

Zužitoval som aj vedomosti získané v predmete URO(Užívateľské rozhrania), kde sme väčšinu času venovali tvorbe UI s dôrazom na prehľadnosť, jednoduchosť a intuitívnosť používania. Vďaka tomuto predmetu som vedel ktoré farebné kombinácie sú nevhodné ba až nečitateľné kvôli fyziognómií ľudského oka. Sústredil som sa na rozloženie jednotlivých prvkov a upútanie pozornosti užívateľa na miesta, ktoré obsahovali dôležité informácie. Vďaka tomuto predmetu bol konečný vzhľad celej aplikácie kompaktnejší a elegantnejší.

## **6 Záver**

Absolvovanie odbornej praxe bolo pre mňa veľkým prínosom. Zdokonalil som si svoje odborné znalosti a zoznámil sa s pracovným prostredím skutočnej firmy. Naučil som sa pracovať v tíme a prezentovať svoju prácu. Vďaka riešeniu reálnych problémov som získal skúsenosti, ktoré by som v prípade riešenia klasickej bakalárskej práce nezískal a navyše som vytvoril niečo skutočne použiteľné.

---

## Použitá literatúra

- [1] InfoMED s.r.o. [online]. 2014 [cit. 2014-04-04] Dostupné z: <http://infomed.sk>.
- [2] Advantage database server info [online]. 2014 [cit. 2014-04-04]. Dostupné z: <http://devzone.advantagedatabase.com/dz/content.aspx?key=1>
- [3] Advantage web platform info [online]. 2014 [cit. 2014-04-04]. Dostupné z: <http://www.sybase.com/products/databasemanagement/advantagedatabaseserver/ADSWebPlatform>
- [4] Android developers [online]. 2014 [cit. 2014-04-04] Dostupné z: <http://developer.android.com/training/index.html>
- [5] IntelliJ IDEA info [online]. [cit. 2014-04-04]. Dostupné z: <http://www.jetbrains.com/idea/>
- [6] Odata info [online]. [cit. 2014-04-04]. Dostupné z: <http://www.odata.org/>
- [7] Odata4j info [online]. [cit. 2014-04-04]. Dostupné z: <http://odata4j.org/>
- [8] JSON info [online]. [cit. 2014-04-04]. Dostupné z: <http://en.wikipedia.org/wiki/JSON>
- [9] Android emulátor info [online]. [cit. 2014-04-04]. Dostupné z: <http://developer.android.com/tools/help/emulator.html>
- [10] Android software development KIT info [online]. [cit. 2014-04-04]. Dostupné z: <http://developer.android.com/sdk/index.html?hl=sk>